INSTITUT NATIONAL DE RECHERCHE
SUR LES TRANSPORTS ET LEUR SÉCURITÉ

Clément FOREST
Mehdi Danech-Pajouh

Problèmes Mal-posés
Un exemple d'application
à la prévision à court-terme

Mars 1998

SOMMAIRE

1. INTRODUCTION 3
2. QU'EST CE QU'UN PROBLÈME MAL-POSÉ ? 4
2.1. NOTIONS DE PROBLÈMES BIEN ET MAL POSÉS 4
2.2. EXEMPLES DE PROBLÈMES MAL-POSÉS 4
3. QU'APPORTENT LES MÉTHODES DE RÉSOLUTION DE PROBLÈMES MAL POSÉS ? ..5
4. EN QUOI LA PRÉVISION À COURT TERME EST-ELLE UN PROBLÈME MAL-POSÉ ? ..5
5. EXEMPLES DE MÉTHODES DE RÉSOLUTION 6
6. CONVERGENCE DE LA MÉTHODE DE RÉGULARISATION DE TIKHONOV 7
6.1. NOTION D'OPÉRATEUR RÉGULARISANT 7
6.2. RÉGULARISATION DE TIKHONOV 7
6.3. MÉTHODE SIMPLE POUR LA DÉTERMINATION DU COEFFICIENT DE RÉGULARISATION 9
7. DESCRIPTION DÉTAILLÉE DE LA MÉTHODE EMPLOYÉE PAR ATLAS 9
8. APPLICATION DE ATLAS À LA PRÉDICTION À COURT-TERME. 10
9. COMPARAISON ATLAS / RÉGRESSION 12
9.1. DESCRIPTION DU SUPPORT DE LA COMPARAISON. 12
9.2. COMPARAISON ENTRE LES ÉVALUATIONS À L'ORDRE 2 ET À L'ORDRE 1 13
9.3. COMPARAISON ENTRE ATLAS ET UNE RÉGRESSION CLASSIQUE 14
9.4. GRAPHIQUES 16
10. ETUDE DU CHOIX DU COEFFICIENT Λ 17
10.1. ERREUR SUR L'ESPACE D'APPRENTISSAGE 17
10.2. ERREUR SUR L'ESPACE DE GÉNÉRALISATION 17
10.3. ERREUR SUR L'ESPACE DE PRÉVISION 18
10.4. CONCLUSION. 18
11. CONCLUSION SUR L'INTÉRÊT DE LA MÉTHODE DE RÉGULARISATION. 20

ANNEXE 1 21
1. OBJET DU LOGICIEL 21
2. MODE D'EMPLOI 21
2.1. FICHIERS D'ENTRÉE 21
2.2. COMMANDES DE L'INTERFACE 23
2.3. FICHIERS DE SORTIE 24
3. REMARQUES DIVERSES SUR L'UTILISATION DE ATLAS 24

ANNEXE 2 26
1. LE FICHIER TEXTE 26
2. LE FICHIER SOURCE .C 26
2.1. DÉCLARATIONS DE VARIABLES 26
2.2. DÉCLARATION DE FONCTION 28
2.3. EXPLOITATION DU FICHIER .C 28

ANNEXE 3 29
1. LA PRÉVISION 29
2. LA MODÉLISATION 29
3. L'INTERPRÉTATION 30
3.1. RÉCUPÉRATION DES RÉSULTATS D'ATLAS 30
3.2. AUTRES OUTILS INTÉRESSANTS 31

ANNEXE 4 35

BIBLIOGRAPHIE 36

1. Introduction

L'objet de ce rapport est d'introduire le lecteur aux méthodes de résolution de problèmes mal-posés et de lui montrer en quoi cette méthode, utilisée ici dans le cadre du logiciel ATLAS de la société SOFRESUD, est plus avantageuse que les méthodes classiques à base de régression. Cette étude s'insère dans le cadre de notre travail de réflexion, demandé par la DSCR, sur une nouvelle génération du dispositif Bison Futé.

On introduira dans un premier temps la notion de problème mal-posé pour ensuite s'intéresser à leurs méthodes de résolution et à ce que celles-ci peuvent apporter à la résolution d'un problème.

On décrira dans ses grandes lignes ce logiciel en s'attardant plus précisément sur la méthode de résolution de Tikhonov qu'il utilise et dont on fournit une ébauche de justification.

Après un exemple des résultats issus de cette approche, on les comparera avec ceux obtenus par une régression classique. On s'intéressera enfin aux méthodes employées par ce logiciel pour fixer de manière automatique la valeur du coefficient de régularisation λ afin de savoir s'il est possible d'améliorer ce choix.

On conclura enfin l'étude sur les avantages que présente la résolution par la méthode des problèmes mal-posés sur d'autres méthodes de résolution plus classiques.

L'annexe 1 propose un mode d'emploi succinct du logiciel ATLAS, de manière à faciliter son utilisation.

L'annexe 2 décrit plus précisément les fichiers fournis en sortie par ce logiciel.

L'annexe 3 expose quelques méthodes d'exploitation du logiciel passant, entre autre, par l'exploitation des fichiers de sortie.

2. Qu'est ce qu'un problème mal-posé ?

2.1. Notions de problèmes bien et mal posés

Résoudre un problème numérique c'est trouver sa solution z à partir des données ou conditions initiales u,

$$z = R(u).$$

Ce problème est dit **correctement posé** s'il possède les propriétés suivantes :

1) pour tout élément $u \in U$ il existe une solution $z \in F$;
2) la solution est définie de façon unique ;
3) le problème est stable sur les espaces (F,U),

c'est-à-dire qu'une variation infinitésimale du second membre ne va pas perturber la solution plus que de manière infinitésimale. Une approximation de la solution est aussi une solution approchée. Ou encore :

$$\forall\, \varepsilon > 0,\ \forall\, u_1, u_2 \in U,\ \forall\, z_1, z_2 \in F,$$
$$\exists\, \delta > 0 \ \text{ tq. } d(u_1,u_2) < \delta \Rightarrow d(z_1, z_2) < \varepsilon ;$$

F et U représentant les espaces d'appartenance de z et u, d représentant indistinctement les distances sur ces deux espaces,

Les conditions 1) et 2) caractérisent sa détermination mathématique, la condition 3) sa détermination physique ainsi que la possibilité d'emploi des méthodes numériques de sa résolution à partir de données approchées.

Les problèmes qui ne possèdent pas ces propriétés sont appelés **problèmes mal-posés**.

2.2. Exemples de problèmes mal-posés

- Un exemple classique de problème mal-posé est la dérivation d'une fonction u(t) connue approximativement.

- On montre que la résolution de l'équation $\int_b^a K(x,s)z(s)ds = u(x)$ est mal-posée. Il s'agit d'une intégrale dite de Fredholm que l'on trouve lorsqu'on s'intéresse au problème inverse de la diffusion de chaleur.

- Si A est une matrice non-inversible, la résolution de Az = u est un problème mal-posé.

3. Qu'apportent les méthodes de résolution de problèmes mal posés ?

La résolution de problèmes mal-posés présente plusieurs difficultés :

- la solution peut être très sensible aux conditions initiales. Celles-ci n'étant la plupart du temps connues qu'avec une précision finie, on observe alors un caractère aléatoire à la solution qui peut paraître insupportable lors d'une utilisation pratique des résultats.
- cette solution peut, du fait de l'incertitude sur les données, ne pas être mathématiquement définie. Comme on veut malgré tout un résultat, il faut rechercher parmi les non-solutions de ce problème celle qui répond le mieux à notre attente : il faut donc arriver à caractériser cette attente de façon plus précise.

Les méthodes de résolution de problèmes mal-posés fournissent à coup sûr une solution unique au problème posé. Celle-ci peut-être un peu moins fidèle que la solution du problème classique mais elle possède la propriété d'être stable, c'est-à-dire d'être moins sensible à de faibles variations des conditions initiales, entraînant par là une plus grande robustesse des résultats.

4. En quoi la prévision à court terme est-elle un problème mal-posé ?

L' INRETS/DART, dans le cadre d'une réflexion menée autour d'une nouvelle génération de Bison Futé, doit se pencher sur les modèles de prévision du trafic routier horaire un an à l'avance, à partir de données uniquement calendaires. L'étude présente s'est effectuée sur le péage de Saint-Arnoud pour lequel on dispose des données du trafic horaire pour chacun des jours de ces dix dernières années. On cherche en fait à réaliser un modèle pour chaque heure de chaque mois (ex : AVRIL 10H, MAI 18H …).

La méthode actuelle consiste à effectuer sur les données correspondantes une régression classique du type

$$\mathbf{Az = u}$$

où u est un vecteur dont chaque élément représente la valeur du trafic pour l'un des jours de l'historique et A est une matrice dont les éléments correspondent aux valeurs des variables explicatives pour chacun de ces jours.

Les résultats sont alors appliqués aux données calendaires du jour désiré de façon à établir la prévision.

On cherche donc à résoudre le problème min J(z) avec $J(z) = ||Az-u||_2$

Plusieurs éléments font en sorte que l'on peut ici considérer ce problème comme mal posé :

La matrice A peut ne pas être de rang plein ou ses colonnes peuvent être fortement corrélées. Cela entraîne la multiplicité des solutions ou une extrême sensibilité de celles-ci par rapport à de faibles variations des données A et u.

Le choix des différentes variables explicatives étant arbitraire et imparfait, les résultats recherchés ne sont qu'indicatifs, et on ne veut pas qu'une variation de quelques véhicules sur l'une ou l'autre des données de l'historique ne vienne perturber de manière sensible les résultats. On ne doit pas exiger d'un tel modèle des précisions d'un ordre supérieur à une cinquantaine de véhicules.

Sur ces points, la méthode de résolution par régression classique n'est pas satisfaisante. Lorsque la matrice A est mal conditionnée, les résultats sont beaucoup trop sensibles à de faible variations des données initiales.

5. Exemples de méthodes de résolution

Les méthodes de résolution évoquées ici sont appelées **méthodes de régularisation**. Ces méthodes consistent à introduire dans le problème une information a priori sur la solution recherchée.

Généralement, au lieu de chercher **min $||Az-u||_2^2$**, ce qui correspond à l'approximation des moindres carrés, on cherche

$$\textbf{min } ||Az-u||_2^2 + \lambda R(z),$$

où R(z) caractérise l'information apportée et où $\lambda > 0$ est un réel appelé coefficient de régularisation et caractérisant l'importance qu'on apporte à cette information. Plus ce coefficient est grand, plus est importante lors de la résolution l'influence du terme en R(z). En pratique, la valeur de ce réel λ est le plus souvent très petite devant les autres grandeurs types du problème.

R(x) peut être de différents types :

- norme L_2 ou L_1, $R(z) = ||z||_2^2$ ou $||z||_1^2$

ex : régularisation de Tikhonov, utilisée typiquement lorsque les matrices sont mal conditionnées ou lorsqu'il existe une infinité de solutions au problème posé.

- opérateur de type dérivé ou son équivalent discret, $R(z) = ||Lz||_2$ ou Δz

ex : utilisé en restitution d'image lorsqu'on veut introduire l'idée d'une corrélation entre plusieurs pixels.

- opérateur de normalisation $R(z) = ||z-z_1||_2$ utilisé lorsque l'on recherche une solution de Az=u qui soit normale à z_1.

La méthode dite de l'entropie consiste à trouver, parmi les solutions vérifiant $||Ax-y||_2^2 < \lambda$ celle qui maximise l'entropie

$$s(z) = -\Sigma x_i \ln(z_i/z^0_i)-(z_i-z^0_i),$$

x^0 étant une approximation initiale de la solution.

Il existe encore beaucoup d'autres méthodes.

La résolution de chacun de ces problèmes fournit une solution approchée de Az=u possédant les propriétés de stabilité recherchées.

6. Convergence de la méthode de régularisation de Tikhonov

6.1. Notion d'opérateur régularisant

On considère l'équation Az = u où A est un opérateur d'un espace F dans un espace U.

Si l'opérateur inverse A^{-1} n'est pas continu et qu'on ne dispose pas de la valeur exacte u_{ex} mais de u_δ vérifiant $d(u_{ex},u_\delta) < \delta$, la solution approchée $z_\delta=A^{-1}u_\delta$ ne peut évidemment pas être considérée comme une approximation de $z_{ex}=A^{-1}u_{ex}$.

Définition : Un opérateur $R(u,\lambda)$ est appelé **opérateur régularisant** pour l'équation Az = u dans le voisinage de $u = u_{ex}$ s'il vérifie les propriétés suivantes :

1) $\exists\ \delta_1 > 0$ tq $R(u,\lambda)$ soit défini sur $\{\lambda > 0\} \times \{u \in U \mid d(u,u_{ex}) < \delta_1\}$,

2) il existe une fonction $\lambda(\delta)$ telle que

$\forall\ \varepsilon > 0, \exists\ \delta \in\ ]\ 0.\delta_1]$ tel que

$$\mathbf{d(u_{ex},u_\delta) < \delta \Rightarrow d(z_{ex},z_\delta) < \varepsilon}$$

avec $z_\delta = R(u_\delta, \lambda(\delta))$.

Cette définition ne suppose pas que l'opérateur $R(u,\lambda)$ soit univoque et il existe en fait une grande diversité d'opérateurs régularisants.

Un tel opérateur R est donc capable de fournir une approximation de z_δ aussi précise que l'on veut, pour peu que l'on dispose d'une précision suffisante sur u_{ex}.

Le problème de recherche d'une solution approchée de Az = u stable vis-à-vis de faibles variations du second membre se réduit donc à :

1) rechercher un opérateur régularisant ;

2) définir le paramètre de régularisation λ.

6.2. Régularisation de Tikhonov

On énonce sans les démontrer les résultats justifiant la méthode de régularisation de Tikhonov. Cette méthode consiste à prendre pour solution du problème $Az=u_{ex}$ la solution de $\min ||Az-u||_2^2 + \lambda||z||_2^2$.

On suppose que l'équation $Az = u_{ex}$ n'admet qu'une seule solution z_{ex}.

6.2.1. Fonction stabilisatrice

Soit $\Omega(z)$ une fonctionnelle positive continue d'un sous ensemble F_1 de F dense sur F.
On dit que Ω est une fonctionnelle stabilisatrice si Ω vérifie :
a) z_{ex} appartient au domaine de définition de Ω ;
b) $\forall\ d > 0$, l'ensemble des z tels que $\Omega(z) \leq d$ est un compact de F_1.

Observons que l'opérateur $z \rightarrow ||z||_2^2$ est bien évidemment stabilisateur.

6.2.2. Existence d'une solution à $\min ||Az-u||_2^2 + \lambda\Omega(z)$

Soit F l'ensemble des solutions possibles de Az = u, et $\Omega(z)$ une fonctionnelle stabilisante définie sur $F_1 \subset F$.

Théorème : Soit A un opérateur continu de F dans U. Quels que soient $u \in U$ et $\lambda > 0$, il existe un élément $z_\lambda \in F_1$ qui minimise la fonctionnelle $M^\lambda(z,u) = ||Az-u||_2^2 + \lambda\Omega(z)$,
c'est-à-dire tel que $\min M^\lambda(z,u) = M^\lambda(z_\lambda,u)$.

Posons $R_1(u,\lambda) = z_\lambda$, ce qui est toujours possible le théorème affirmant l'existence de z_λ.

6.2.3. Caractère régularisant de R_1

Théorème : Soit z_{ex} solution de $Az = u_{ex}$ ($Az_{ex} = u_{ex}$). Alors
$\forall\ \varepsilon > 0$
$\forall\ \beta_1, \beta_2$ deux fonctions continues, non négatives et non décroissantes sur $[0 \ldots \delta_1]$ telles que $\delta^2/\beta_1(\delta) \leq \beta_2(\delta)$.
$\exists\ \delta 0 \leq \delta_1$ tq $\forall\ \delta \leq \delta 0$, $\forall\ \lambda \in [\delta^2/\beta_1(\delta) \ldots \beta_2(\delta)]$

$$\mathbf{d(u_{ex},u) \leq \delta \Rightarrow d(z_{ex},z_\lambda) \leq \varepsilon}$$

avec $z_\lambda = R_1(u,\lambda)$

L'opérateur R_1 est donc bien régularisant au voisinage de u_{ex}. En fait, R_1 est régularisant pour tout $u \in U$.

En rappelant le caractère stabilisateur de $z \rightarrow ||z||_2^2$, on a bien la validation de la méthode de régularisation de Tikhonov.

6.3. ***Méthode simple pour la détermination du coefficient de régularisation***

Une difficulté des méthodes de régularisation réside dans le choix de la valeur du paramètre λ. Il existe une grande variété de méthodes permettant d'optimiser ce choix reposant sur une étude attentive du problème posé.

Il existe toutefois une méthode empirique très simple à mettre en œuvre, appelée (en anglais) Generalized cross-validation (GCV).

L'idée consiste à mettre de côté l'une des données y_i du problème et à considérer que la valeur optimale de λ conduit à une bonne approximation de cet y_i. Le λ choisit sera alors celui qui optimise cette approximation.

7. Description détaillée de la méthode employée par ATLAS

Le logiciel ATLAS repose sur la méthode de régularisation de Tikhonov. Il cherche donc à minimiser la fonctionnelle

$$J(z) = ||Az-u||_2^2 + \lambda||z||_2^2$$

où A et u représentent un ensemble de données historiques que l'on fourni au programme et où λ est un réel fourni par l'utilisateur. ATLAS peut cependant déterminer automatiquement la constante λ idéale à l'aide d'une méthode similaire à la méthode GCV évoquée plus haut et qui est plus précisément décrite au chapitre 10.

Les données historiques que l'on fournit au logiciel sont divisées en trois ensembles :

- l'ensemble d'apprentissage : il comprend les données sur lesquelles seront effectuées les calculs et qui constitue en fait la matrice A .
- l'ensemble de généralisation : constitué de données historiques qui auraient pu a priori faire partie de l'ensemble d'apprentissage mais que l'on a mises à part et qui servent à orienter les algorithmes de convergence internes. Pour une efficacité optimale, cet ensemble doit être environ 4 à 5 fois plus petit que le précédent.
- l'ensemble de prévision est celui sur lequel est effectué la prévision.

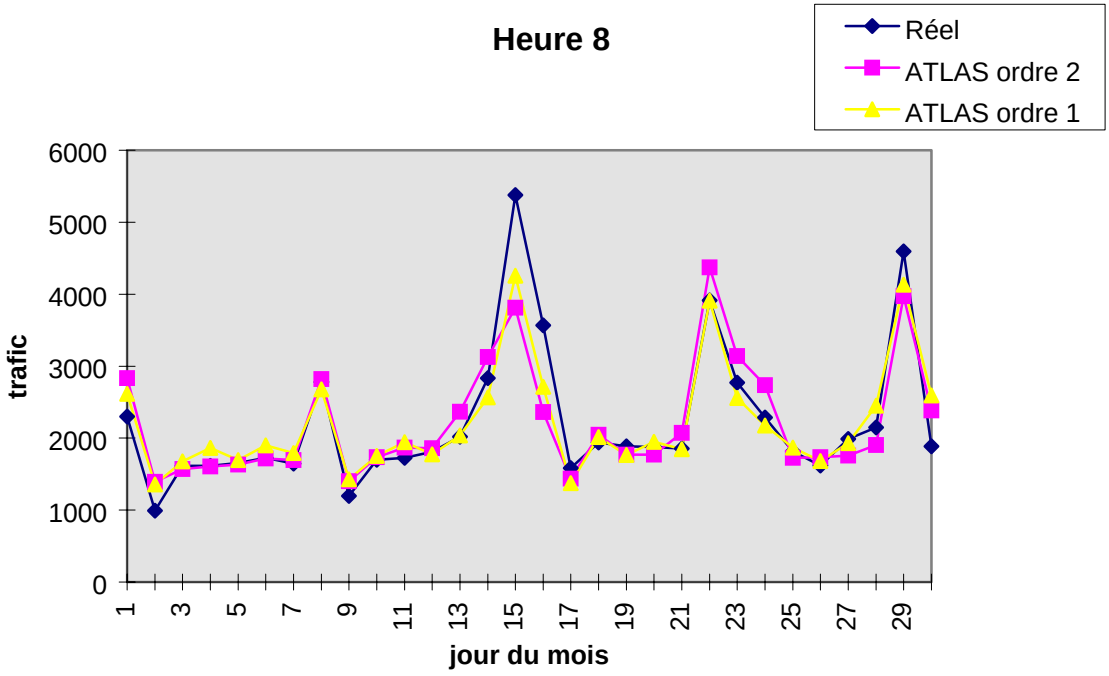
L'un des travaux de l'utilisateur consiste donc à séparer les deux premiers ensembles. Cette séparation ne nécessite pas de précautions particulières si ce n'est que l'ensemble de généralisation doit être le plus possible représentatif du comportement du système.

ATLAS possède aussi d'autres caractéristiques le distinguant un peu plus d'un logiciel de régression classique :

- la capacité d'effectuer la recherche en considérant non seulement les variables explicatives proposées par l'utilisateur, mais aussi les combinaisons de celles-ci, jusqu'à un degré choisi arbitrairement par l'utilisateur, ce qui permet une modélisation plus fine.
- la séparation des données historiques en deux ensembles, un ensemble d'apprentissage sur lequel sont effectués les calculs, et un ensemble de généralisation qui sert à optimiser et à orienter la convergence de l'algorithme de minimisation de la fonctionnelle.

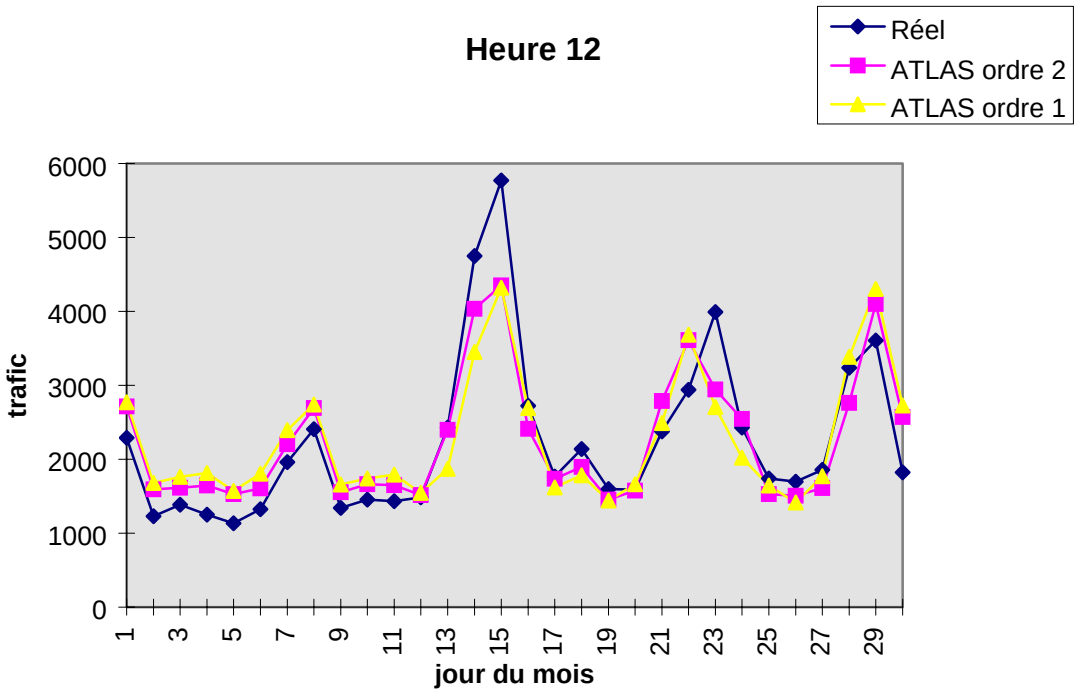
8. Application de ATLAS à la prédiction à court-terme.

Voici quelques exemples des prédictions (horizon un an) effectuées par ATLAS à l'ordre 2 et à l'ordre 1 pour le mois d'avril 1995 aux heures 8, 12, 15 et 20. On a indiqué de plus les erreurs L_2 sur chacun des trois espaces : apprentissage, généralisation, et prévision. Cette erreur correspond à la racine carrée de la moyenne des erreurs quadratiques.



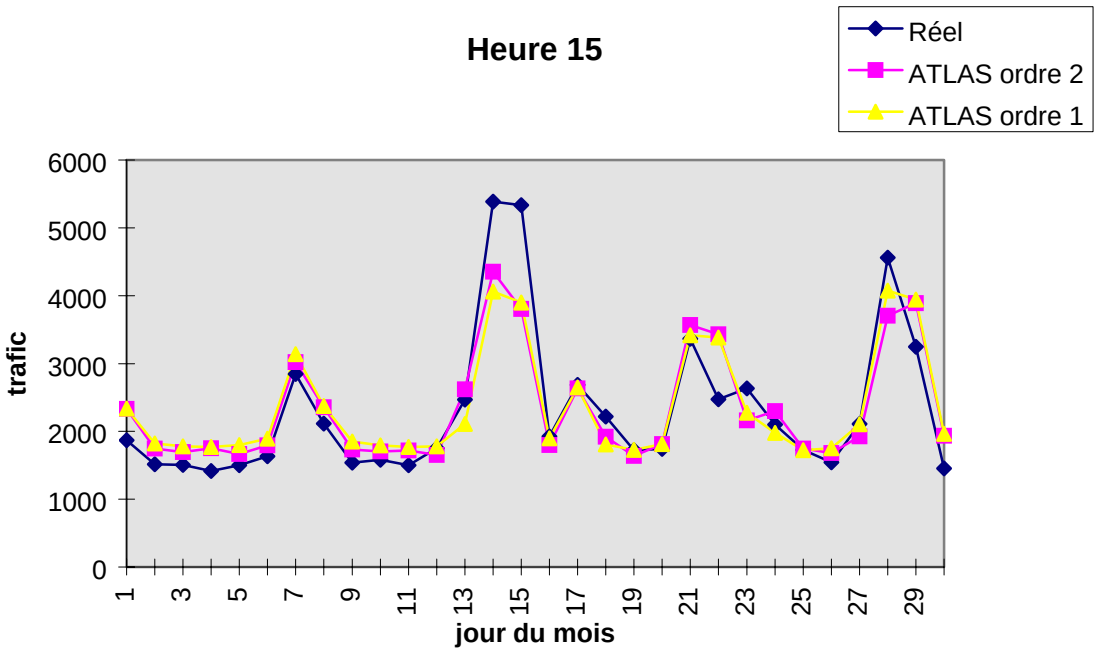
Erreur L_2 :

Espace	Ordre 1	Ordre 2
apprentissage	347	246
généralisation	304	351
prévision	339	449



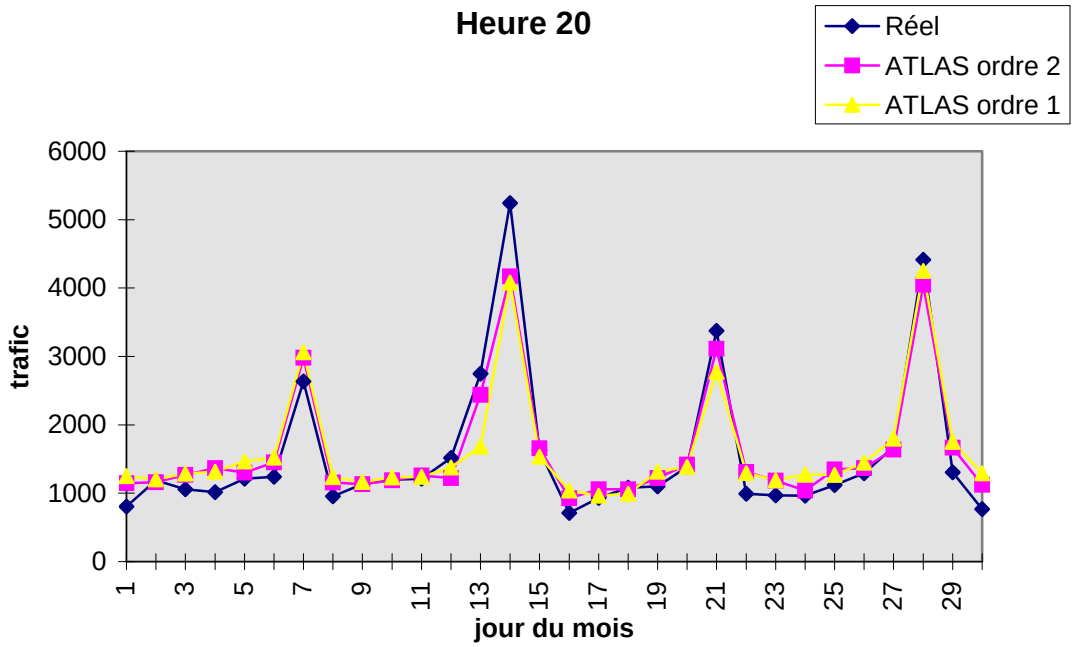
Erreur L_2 :

Espace	Ordre 1	Ordre 2
apprentissage	364	218
généralisation	351	327
prévision	577	470



Erreur L_2 :

Espace	Ordre 1	Ordre 2
apprentissage	277	198
généralisation	309	355
prévision	488	478



Erreur L_2 :

Espace	Ordre 1	Ordre 2
apprentissage	228	140
généralisation	325	349
prévision	390	293

9. Comparaison ATLAS / Régression

On compare ici différentes utilisations d'ATLAS (Ordre 1 et Ordre 2) entre eux puis avec une régression classique.

9.1. Description du support de la comparaison.

Pour la comparaison, on a cherché à créer un modèle qui détermine, pour une heure donnée en avril, le trafic routier un an à l'avance au péage de Saint-Arnoud, et cela uniquement à partir de données calendaires. On a donc créé 24 modèles différents : avril_H0_, avril_H1_,…, avril_H23_. Les modèles effectivement utilisés lors de la comparaison correspondent aux heures 4, 6, 8, 10, 12, 15, 18, 20 et 23.

Les variables utilisées pour la description, décrites Annexe 4, sont au nombre de 29 et sont principalement des variables binaires servant à repérer des événements calendaires (débuts de week-end, de congé scolaire …).

Pour ATLAS, on a séparé pour chaque heure les données en trois paquets :

- les années 1986 à 1991 constituent l'espace d'apprentissage,
- les années 1992 à 1994 constituent l'espace de généralisation,
- l'année 1995 constitue l'espace sur lequel sera effectuée la prévision.

Pour la régression, on a fourni une variable explicative supplémentaire, constante, à la régression, de manière à ce que le modèle généré puisse ne pas passer par l'origine. Ce modèle est créé à partir des années 1986 à 1994, l'année 1995 servant à comparer les résultats.

9.2. Comparaison entre les évaluations à l'ordre 2 et à l'ordre 1

Une augmentation de la valeur de l'ordre augmente de façon importante le nombre de variables explicatives de façon à rechercher une plus grande finesse du modèle.

Pour les fichiers d'AVRIL 95, ce nombre passe de 29 variables pour l'ordre 1 à 175 variables pour l'ordre 2, ce qui est maximal compte tenu du fichier d'apprentissage qui contient 178 éléments.

On a effectué une comparaison de la précision des prévisions obtenues pour l'ordre 1 et pour l'ordre 2.

On observe :

Pour les heures 4 et 23, le modèle obtenu par l'ordre 2 est beaucoup plus précis que celui à l'ordre 1, les différences de pourcentage d'erreur étant respectivement de 15% et 20%.

Pour les heures 6, 8, 10, et 18 le modèle à l'ordre 1 est plus précis que celui à l'ordre 2 mais les différences de 0,3%, 2,5%, 1,6% et 1,7% peuvent être considérées comme non significatives compte tenu du faible nombre d'individus pris en compte pour le calcul de ces erreurs (30 pour le mois d'Avril 95).

Les heures 12, 15 et 20 montrent une meilleure précision du modèle à l'ordre 2 mais on peut encore considérer les différences de 4,6% 1,1% et 5,7% comme encore non significatives.

L'erreur du modèle à l'ordre 1 varie entre 15% et 46%.
L'erreur du modèle à l'ordre 2 varie entre 15% et 31%.

En conclusion :

Lorsque le modèle à l'ordre 2 est plus précis que celui à l'ordre 1, il l'est souvent de manière très significative. Dans les autres cas, on peut considérer ces deux modèles comme équivalents.

Ce modèle est de plus beaucoup plus régulier dans ses prévisions que le modèle à l'ordre 1.

L'ordre 2 apparaît donc comme plus intéressant à exploiter. Remarquons l'importance du nombre de variables explicatives qui doit être inférieur au nombre d'éléments du fichier

d'apprentissage pour une modélisation optimum. Si ce nombre de variables explicatives devient trop grand (lorsqu'on augmente le degré ou le nombre de variables d'entrée), les erreurs à la prévision augmentent de manière significative.

9.3. Comparaison entre ATLAS et une régression classique

9.3.1. Avec ATLAS à l'ordre 1

On compare d'abord la régression classique avec ATLAS à l'ordre 1. Le nombre de variables explicatives restant le même, la seule différence entre ces deux méthodes réside dans la technique de modélisation : régression pour l'une et régularisation pour l'autre.

L'apport majeur des techniques de régularisation est d'apporter une plus grande robustesse au modèle généré.

Le choix de la valeur du coefficient de régularisation λ a été laissé à l'ordinateur. Celui-ci choisit à l'intérieur d'un intervalle donné, ici [0,100], la valeur de λ qui minimise l'erreur du modèle lorsqu'il est appliqué sur l'espace de généralisation[1].

On observe :

- pour les heures 6 et 15, la régularisation fournit une prévision beaucoup plus précise qu'une régression normale. Les différences entre les taux d'erreurs étant respectivement de 11% et 19%.
- pour les heures 4, 8, 10, 12 et 18, la régularisation est plus précise que la régression classique mais les différences de taux d'erreur ne sont pas significatives : 2%, 3%, 2%, 1,0% et 0%.
- à 20 heures, la régression est un peu plus précise que la régularisation, mais la différence d'erreur n'est pas significative : 1%.
- à 23 heures, la régression est plus précise que la régularisation et la différence des taux d'erreur est de 6%. On peut se poser la question quant au caractère significatif de cette différence.

Les variations du taux d'erreur de la régularisation et de la régression apparaissent comme équivalents, les erreurs de la méthode de régularisation montant cependant moins souvent au dessus de 30% que la régression classique (2 fois contre 4).

En conclusion :

Sauf pour l'heure 23 où cela est discutable, la méthode de régularisation est, soit plus précise, soit équivalente à une régression classique. Son usage est donc plus intéressant.

[1] Le chapitre 10 fournit plus de précision sur cette détermination.

9.3.2.Avec ATLAS à l'ordre 2

Une utilisation d'ATLAS à l'ordre 2 combine deux modifications par rapport à une méthode de régression classique :

- une recherche de la solution à un plus haut degré que pour la régression simple (ici un polynôme à l'ordre 2 contre un ordre 1 pour la régression simple)
- l'utilisation de la méthode de régularisation pour réaliser l'approximation.

L'ordre 2 correspond à l'utilisation courante d'ATLAS.
Le choix de la valeur de λ est laissé à l'ordinateur.

On observe :

- pour les heures 4, 6, 15 et 23 la régularisation est plus précis que le régression classique, et ce de manière très significative. Les différences de taux d'erreur sont de 18%, 11%, 20% et 14%.
- pour les heures 10, 12 et 20, elle reste encore le plus précis, mais de manière moins significative. Les différences de taux d'erreurs sont de 1%, 5% et 5%.
- pour les heures 8 et 18, la régression classique est la plus précise mais les différences de 0% et 1% ne sont pas significatives.

Le taux d'erreur de la régression varie entre 48% et 15%. Celui de la régularisation varie entre 30% et 15% et est moins souvent proche de ses valeurs maximales (2 fois contre 4).

En conclusion :

Pour la moitié des heures considérées, ATLAS à l'ordre 2 est significativement plus précis qu'une régression classique. Pour les autres heures, le logiciel est soit meilleur, mais de manière moins significative, soit tout à fait équivalent à la régression classique.

Ce modèle reste de plus beaucoup plus régulier dans ses taux d'erreur qu'une régression classique.

L'utilisation à l'ordre 2 semble donc plus avantageuse. La précision est souvent meilleure, au pire équivalente, mais en tout cas beaucoup plus stable que celle d'une régression classique. Cette méthode de régularisation un peu modifiée se montre donc très intéressante.

9.4. GRAPHIQUES

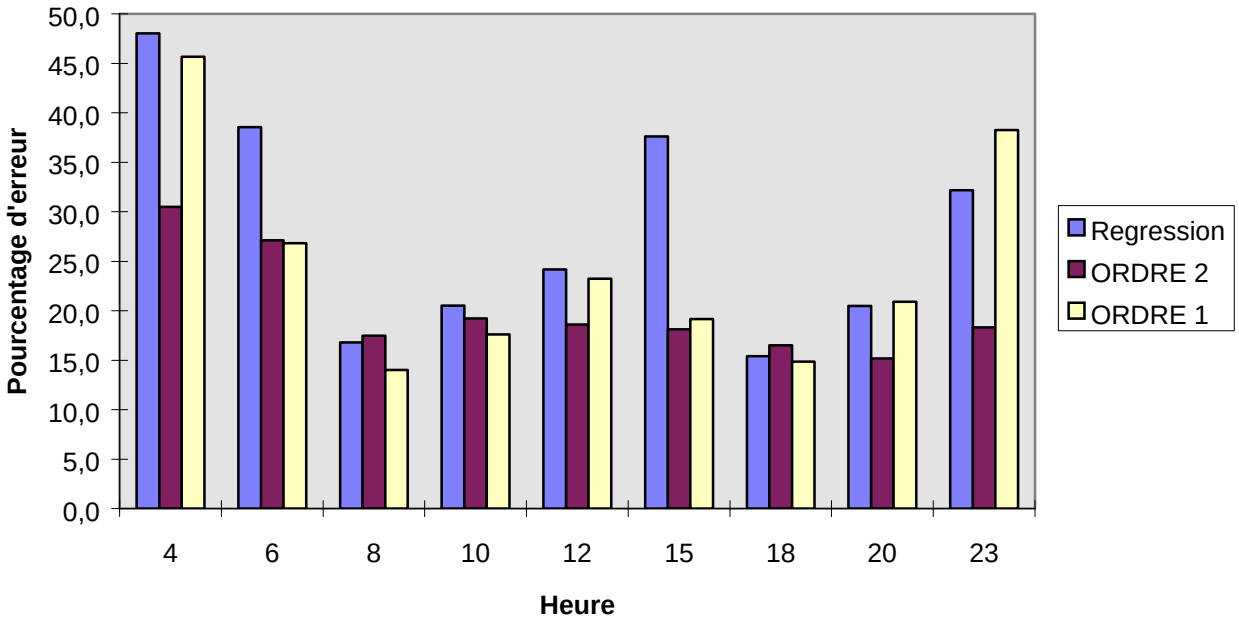
COMPARAISON ATLAS/REGRESSION

AVRIL

Pourcentage d'erreur en fonction de l'heure et de la méthode employée.

HEURE	Régression	ORDRE 2	ORDRE 1
4	48	30	46
6	38	27	27
8	17	17	14
10	20	20	18
12	24	19	23
15	38	18	19
18	15	16	15
20	20	15	21
23	32	18	38

Le pourcentage d'erreur est calculé comme suit :
(erreur quadratique de la méthode) / (nombre réel total de véhicules)*100



10. Etude du choix du coefficient λ

Pour la détermination du modèle A*z = u, le logiciel cherche à minimiser une fonctionnelle de la forme $J_\lambda(z) = ||Az-u||^2 + \lambda||z||^2$. Comme on l'a vu, cette minimisation diffère d'une régression classique par la présence du terme en λ. Ce terme, permettant d'assurer l'existence d'un minimum pour la fonctionnelle ainsi que sa stabilité, caractérise l'éloignement que l'on est prêt à tolérer entre le modèle cherché et celui qui aurait été déterminé par une régression classique.

Un grande valeur de λ est théoriquement souhaitable dans le cas d'un système très chaotique, une valeur faible est souhaitable lorsque le système étudié est très régulier.

Le programme peut soit utiliser la valeur de λ souhaitée par l'utilisateur, soit rechercher la valeur idéale de ce paramètre à l'intérieur d'un intervalle désigné.

Par une étude, pour différentes valeurs de λ, des erreurs du modèle déterminé par le logiciel par rapport à chacun des trois espaces : apprentissage, généralisation et prévision, on a cherché à savoir de quelle manière il déterminait la valeur idéale de λ et si il existait un moyen d'améliorer ce choix.

Rq : on n'a pas étudié ces erreurs pour une valeur de λ égale à 0 car ATLAS n'étant pas prévu pour un tel fonctionnement, les résultats obtenus sont sans signification.

10.1. Erreur sur l'espace d'apprentissage

Pour les cinq heures étudiées, la courbe d'erreur sur l'espace d'apprentissage présente des aspects tout à fait similaires. Cette courbe croît avec λ et cette croissance est de plus en plus forte quand λ s'approche de 0.

Cela semble en cohérence avec le fait que λ caractérise l'éloignement entre le modèle recherché et celui minimisant les moindres carrés.

10.2. Erreur sur l'espace de généralisation

Pour les cinq heures étudiées, la courbe d'erreur sur l'espace de généralisation présente des aspects relativement semblables.

D'une valeur assez élevée aux alentours de 0, l'erreur décroît fortement pour atteindre un minimum correspondant à des valeurs de λ comprises entre 1 et 5. Cette erreur augmente alors légèrement avec λ.

A part pour l'heure 0, cette erreur est toujours supérieure à l'erreur sur l'espace d'apprentissage.

On remarque que, pour chaque heure étudiée, la valeur du λ optimal calculée par l'ordinateur correspond au minimum de cette courbe d'erreur sur l'espace de généralisation. On en déduit que, lorsqu'on lui laisse le choix, le logiciel recherche, parmi les valeurs de λ contenues dans l'intervalle de recherche, celle qui minimise l'erreur sur l'espace de généralisation.

10.3.Erreur sur l'espace de prévision

On aimerait trouver un moyen de déterminer λ minimisant l'erreur sur cet espace de prévision.

Les courbes d'erreur sur l'espace d'apprentissage en fonction de λ sont de trois types :

- courbes décroissantes (ex : Heures 0 et 2).
- courbes croissantes (ex : Heure 6).
- courbe décroissante puis croissante (ex : Heures 8, 10, 15 et 20).

Cette courbe peut croiser la courbe d'erreur sur l'espace de généralisation (Heures 0, 6, 8, 10 et 15). Elle peut aussi être située en dessus (Heure 2) ou en dessous (Heure 20).

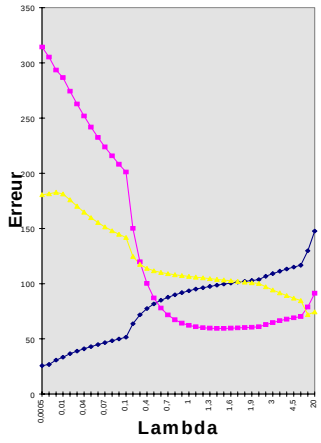
Ne pouvant connaître d'avance aucune caractéristique de cette courbe, on est contraint de se rabattre pour λ sur le choix de l'ordinateur, c'est-à-dire celui qui minimise l'erreur sur l'espace de généralisation.

10.4.Conclusion.

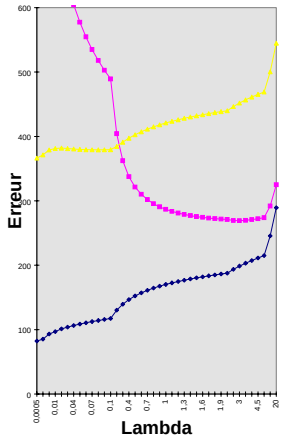
Lorsqu'on demande à ATLAS de déterminer tout seul la valeur idéale de λ, la valeur fournie est celle pour laquelle la valeur de l'erreur sur l'espace de généralisation est minimum. Compte tenu de l'information disponible sur l'espace de prévision et du fait que l'espace de généralisation est constitué d'événements potentiels, cette valeur paraît comme la meilleure à utiliser.

Erreur en fonction de λ pour chacun des trois espaces

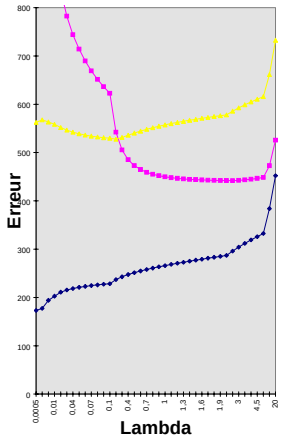
Heure 0



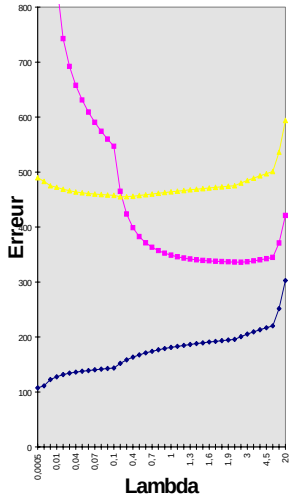
Heure 6



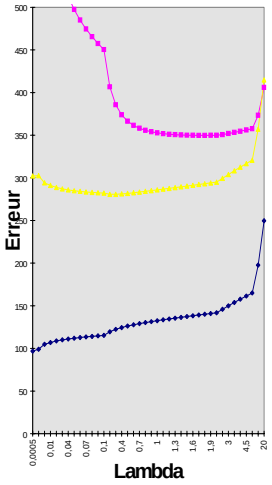
Heure 10



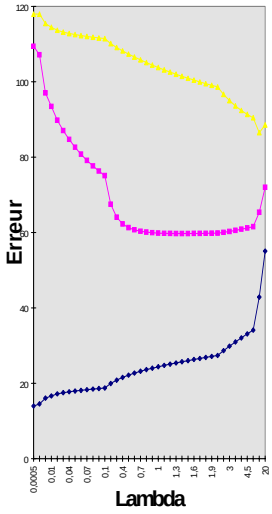
Heure 15



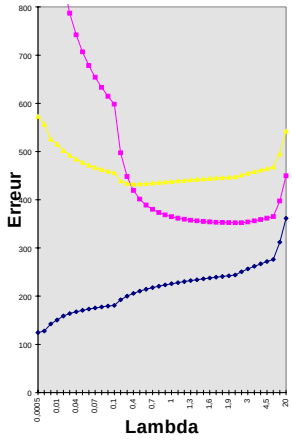
Heure 20

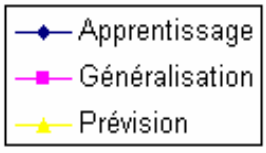


Heure 2



Heure 8





11. Conclusion sur l’intérêt de la méthode de régularisation.

Par rapport aux régressions classiques, les méthodes de régularisation sont plus précises et présentent de plus l'énorme avantage de la robustesse des résultats ; les variations de ceux-ci sont en effet peu sensibles à de faibles fluctuations des données du problème. Le gain de précision de cette approche est encore plus significatif lorsqu'elle est combinée avec une augmentation (mesurée) des variables explicatives.

Ces méthodes présentent cependant une difficulté majeure qui décourage bien souvent sa mise en œuvre : elles nécessitent en effet la détermination par l'utilisateur d'un coefficient λ dit de régularisation et dont dépend la justesse des résultats. La démarche du logiciel ATLAS, consistant en une séparation des données d'étude en deux blocs dont l'un sert de garde-fou, est satisfaisante et simple à mettre en application.

Pour des problèmes, tels que celui de la prévision à court terme, pouvant être considérés comme des problèmes mal posés, les solutions à base de méthodes de régularisation apparaissent pratiques, performantes et qu'il serait intéressant de considérer.

ANNEXE 1
Mode d'emploi

1. Objet du logiciel

ATLAS est un logiciel qui extrapole un jeu de données (variables explicatives + grandeur à apprendre) **pour essayer d'en établir un modèle optimal sous la forme d'un polynôme de variables explicatives**. Il est laissé à l'utilisateur la possibilité de déterminer la précision du modèle en fixant le degré du polynôme généré (commande ORDER), ainsi que, par le choix d'une constante λ dite **constante de régularisation** (commande LAMBDA), sa robustesse c'est-à-dire sa sensibilité à de petites variations du jeu de données fourni.

ATLAS, simultanément, applique ce modèle à un jeu de variable explicatives. C'est sa fonction de **prévision**.

2. Mode d'emploi

2.1. Fichiers d'entrée

L'utilisateur doit fournir trois jeux de données au logiciel, constituant les espaces d'apprentissage, de généralisation et de prévision.

C'est à partir des données contenues dans l'espace d'apprentissage que le logiciel va créer le modèle optimal.

L'espace de généralisation sert à contrôler le sur-apprentissage, c'est-à-dire la trop grande spécificité du modèle obtenu par rapport aux données fournies, ou encore sa robustesse.

Pour des résultats optima, le fichier d'apprentissage doit être environ cinq fois plus grand que le fichier de généralisation, de plus, les données qu'ils contiennent doivent être distinctes.

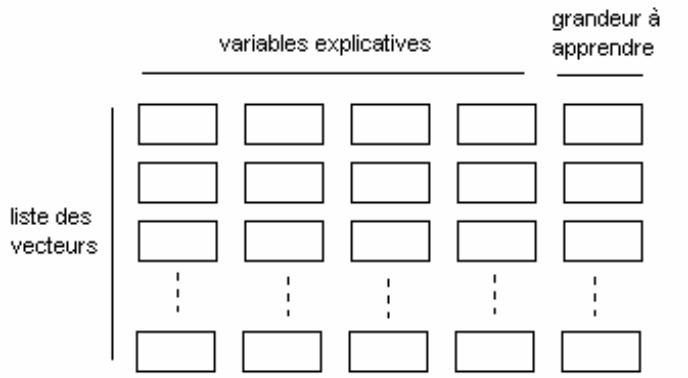
Le fichier de prévision sera celui sur lequel sera appliqué le modèle calculé par ATLAS

2.1.1. Format des fichiers

Les fichiers fournis devront se présenter sous la forme de fichiers textes (*.txt). Chacune des lignes des fichiers correspond à une observation. Elle est composée d'une liste des variables explicatives, séparées les unes des autres par des tabulations, et est terminée par la valeur de la grandeur à expliquer.

Pour le fichier de prévision (*FS), **la présence de cette valeur est nécessaire au bon fonctionnement du programme** mais n'est pas utilisée (elle peut permettre après coup d'effectuer des comparaisons prévision / réalité) et on peut très bien la remplir de manière quelconque (ex : avec des 0).

A noter que l'on ne peut se passer de ce fichier de prévision. ATLAS à besoin de sa présence même s'il n'est composé que d'un vecteur nul.



De tels fichiers peuvent être créés à partir d'Exel ou de Word en utilisant la commande **Fichier … Enregistrer sous … Type de fichiers … Texte seulement**, ou encore à l'aide de n'importe quel éditeur de texte.

2.1.2. Conventions sur le nom des fichiers d'entrée

Les noms des fichiers devront être de la forme :

- <Préfixe>ts pour le fichier d'apprentissage,
- <Préfixe>gs pour le fichier de généralisation,
- <Préfixe>fs pour le fichier de prévision.

Il suffit alors d'indiquer au programme le préfixe commun aux trois noms par la commande **data**.

Ex : **data avril_H10_** indique qu'on travaillera sur avril_H10_ts.txt, avril_H10_gs.txt et avril_H10_FS.txt.

Les commandes **ts**, **gs** et **fs** permettent cependant de spécifier pour chaque phase et de manière indépendante le fichier que l'on désire utiliser.

Ex : **fs avril_H11_fs.txt** indique que la prévision se fera sur avril_H11_fs.txt et non plus sur avril_H10_fs.txt.

2.2. Commandes de l'interface

2.2.1. Obtenir de l'aide

« help » et **« ? »** donnent accès à la liste des commandes disponibles et présente brièvement leur effet.

« print » affiche quelle est la valeur courante de chaque paramètre de la modélisation.

2.2.2. Quitter ATLAS

« quit », **« exit »** et **« . »** mettent fin à la session ATLAS.

2.2.3. Spécifier les fichiers d'entrée

« data » suivi d'un nom de préfixe permet de spécifier en une seule opération le nom des trois fichiers d'entrée à condition qu'ils se nomment : <Préfixe>TS.txt, <Préfixe>GS.txt et <Préfixe>FS.txt. (voir plus haut).

« ts », **« gs »** et **« fs »** suivis d'un nom de fichier permettent de spécifier les fichiers utilisés dans les phases d'apprentissage, de généralisation et de prévision.

2.2.4. Spécifier le fichier de sortie

« ps » suivi d'un préfixe permet de spécifier le nom des fichiers de sortie. Ces fichiers se nomment <Préfixe>_<ordre du modèle>.txt et <Préfixe>_<ordre du modèle>.C (voir plus loin).

2.2.5. Spécifier l'ordre du modèle

« order » permet de fixer l'ordre du modèle, c'est-à-dire le degré du polynôme recherché.

« ordermin » et **« ordermax »** indiquent à ATLAS les bornes à l'intérieur desquelles il recherchera l'ordre optimum.

On utilise le plus fréquemment l'ordre 2 car il apporte une précision sensiblement meilleure que l'ordre 1 tout en conservant des temps de calculs supportable.

Augmenter l'ordre entraîne une croissance exponentielle des variables explicatives possible. Il faut donc s'assurer que ce nombre reste inférieur au nombre d'éléments de l'espace d'apprentissage et que les temps de calculs ne deviennent pas trop grands.

2.2.6. Spécifier le paramètre de régularisation

« lambda » permet de fixer la valeur du paramètre de régularisation, c'est-à-dire la sensibilité du modèle trouvé par rapport aux données fournies en entrée.

« lambdamin » et **« lambdamax »** **« ordermin »** et **« ordermax »** indiquent à ATLAS les bornes à l'intérieur desquelles il recherchera le paramètre de régularisation optimum.

2.2.7. Lancer l'apprentissage et réaliser la prévision

« go » permet de lancer l'apprentissage puis de réaliser la prévision. A la fin de cette opération, les erreurs L_1,L_2 et L_∞, obtenues en apprentissage, en régularisation et en prévision sont mesurées et affichées à l'écran.

2.3. Fichiers de sortie

ATLAS fournit en sortie deux fichiers, un fichier texte et un source en C, appelés respectivement <Préfixe>_<ordre du modèle>.txt et <Préfixe>_<ordre du modèle>.C.

2.3.1. Le fichier texte

Ce fichier est une recopie du fichier de prévision auquel est a ajouté une dernière colonne constituée des valeurs que le modèle a prévues pour chacun des vecteurs constituant le fichier de prévision.

2.3.2. Le source en C

Il s'agit d'un source directement exploitable dans un programme en C et qui contient les résultats de la modélisation.

3. Remarques diverses sur l'utilisation de ATLAS

Ne pas terminer une session ATLAS autrement qu'avec les commandes **quit** ou **exit.** En effet, si on sort de manière trop brutale le programme refuse de se relancer à nouveau et informe : « plus de licence ». Il est alors nécessaire d'arrêter puis de relancer le serveur. Ce qui s'effectue de la manière suivante :

Démarrer … Paramètres … Panneau de configuration … ADMS SOFRESUD … Stop, puis **Launch**, puis **OK**.

Ne pas imposer 0 comme valeur de λ. Cela cause des problèmes d'exécution du programme et ne fournit aucun résultat intéressant.

ATLAS doit toujours être accompagné du Fichier Formula.C qui permet la génération de l'un des fichiers de sortie. Son absence dans le répertoire entraîne une erreur d'exécution.

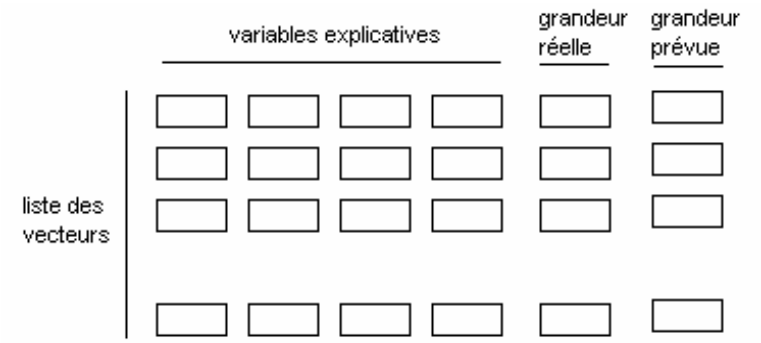
ANNEXE 2
Description des fichiers de sortie

ATLAS fournit en sortie deux fichiers différents, un fichier texte nommé <Préfixe>_<Ordre de modèle>.txt et un fichier contenant un source en C appelé <Préfixe>_<Ordre du modèle>.C

Voici une description détaillée du contenu de ces deux fichiers.

1. Le fichier texte

Ce fichier est une recopie du fichier de prévision (*fs.txt) auquel est a ajouté une dernière colonne constituée des valeurs que ATLAS a prévues pour chacun des vecteurs constituant le fichier de prévision. Il possède la forme suivante :



Les premières colonnes sont identiques à celles du fichier de prévision (*.fs), et la dernière contient le résultat de la prévision.

2. Le fichier source .C

C'est un fichier susceptible d'être inclus dans n'importe quel programme C. Il est composé de deux parties :

- une partie composée de déclarations de variables.
- une partie consistant en une déclaration de fonction.

Le squelette de ce fichier est contenu dans le fichier Formula.

2.1. Déclarations de variables

Le fichier contient un certain nombre de déclarations de variables servant à retrouver le polynôme de modélisation.

2.1.1. Déclarations de constantes

Ces constantes sont définies à l'aide d'un instruction #define. Elles sont donc utilisables par tout programme situé dans le même bloc.

C_NonDefini : Vaut -1. Lorsque, dans la suite des déclarations, un terme n'est pas défini, on met sa valeur à -1. Cette constante sert à accélérer la fonction de calcul de la prévision.

C_Degre : Degré du polynôme de modélisation. Ordre de la recherche.

C_NombreVE : Nombre de variables explicatives. Pour effectuer une modélisation à un ordre plus grand que 1, ATLAS considère tous les monômes de degré inférieur ou égal à l'ordre de la modélisation. *C_NombreVE* représente le nombre de ces variables.

C_NombreNonNull : Parmi ces monômes, un certain nombre sont intrinsèquement nuls ou constants et certains peuvent être deux à deux proportionnels. ATLAS repère ces monômes et ne s'en servira plus dans la suite de ses calculs. *C_NombreNonNull* représente le nombre de monômes n'ayant pas étés supprimés et servant effectivement dans le polynôme.

Rq : ATLAS ne repère pas les colonnes colinéaires et se contente de celles qui sont deux à deux effectivement proportionnelles.

2.1.2. Déclaration de tableaux

ecartType : Ce tableau de dimension *C_NombreVE*+1 contient, pour chacun des monômes, la valeur de l'écart type. Le dernier élément du tableau est égal à l'écart type de la grandeur à expliquer.

moyenne : Ce tableau de dimension *C_NombreVE+1* contient la valeur moyenne de chacun des monômes. Le dernier élément est égal à la valeur moyenne de la grandeur à expliquer.

Rq : Valeur moyenne et écart type sont calculés à partir de l'ensemble des données contenues dans le fichier d'apprentissage.

vecteurW : Ce tableau de dimension *C_NombreNonNull*+1 contient, pour chacun des monômes prenant effectivement part au polynôme de modélisation, la valeur du coefficient W dans la formule :

$$Y = \hat{Y} + \sigma_y * \sum_{i=0}^{C_NombreNonNull-1} W_i \frac{X_i - \hat{X}_i}{\sigma_i}$$

Le dernier élément de ce tableau est toujours mis égal à zéro.

vecteurNonNull : Ce tableau de dimension *C_NombreNonNull* contient la liste des indices des monômes n'étant pas considérés comme nuls.

Le coefficient W du $i^{ème}$ monôme (numérotés de 0 à *C_NombreVE*) est l'élément *vecteurNonNull*[i] du tableau *vecteurW*.

transformation : Ce tableau de dimension (*C_NombreVE*+1)×(*C_Degre*+1) explicite la signification de « $i^{ème}$ monôme ». En gros, *transformation* établit la liaison entre les variables d'entrée (celles dont les valeurs sont fournies par l'utilisateur) et les variables explicatives. La $i^{ème}$ variable explicative est le produit des variable d'entrée d'indices *transformation*[i][n], n variant de 0 à *C_Degre*-1, en appelant variable d'entrée d'indice *C_NonDefini* (-1) la constante 1.

transformation[i][.]=1 ; 3 ; 7 ; -1 ; -1.... signifie que la $i^{ème}$ variable explicative X_i est égale au produit des variables d'entrée x_1, x_3 et x_7

$$X_i=x_1*x_3*x_7$$

Rq : Le tableau *transformation* est bordé en haut et à droite par des termes de valeur *C_NonDefini* (-1). Ces termes servent lors du calcul de la prévision.

2.2. Déclaration de fonction

Est ensuite déclarée une fonction appelée *formula*<Ordre du modèle> qui permet de calculer une approximation.

Cette fonction nécessite en entrée un vecteur de type *float* et correspondant aux valeurs des variables d'entrée pour la prévision désirée et fournit en sortie un réel de type *float* égal à la valeur de cette prévision.

2.3. Exploitation du fichier .C

Ce fichier est utilisable facilement dans n'importe quel programme C via l'instruction *#include* du pré-compilateur.

Il suffit d'insérer au début du source l'instruction : **#include <NomDuFichier>.C** La fonction et toutes les variables qui y sont déclarées sont alors utilisables dans la suite du programme.

ANNEXE 3

EXPLOITATION DES RESULTATS

On peut exploiter ATLAS de trois manières différentes :

- pour la prévision,
- la modélisation,
- pour l'interprétation.

1. La prévision

La prévision reste l'usage principal du logiciel ATLAS, ce pour quoi il a été conçu. Réaliser une prévision correspond à l'exploitation du fichier texte de sortie. Pour réaliser ce fichier, ATLAS recopie le fichier de prévision (*fs.txt) auquel il ajoute une colonne contenant la prévision obtenue pour chacun des vecteurs composant le fichier de prévision. Une lecture de la dernière colonne de ce fichier permet donc de connaître les résultats de la prévision.

2. La modélisation

Pour effectuer une prévision, ATLAS considère les fichiers d'apprentissage et de généralisation (ts et gs) et détermine un polynôme qu'il applique ensuite au fichier de prévision (fs). Ces deux étapes s'effectuent l'une après l'autre lors d'une utilisation normale du logiciel et il n'est pas possible de les dissocier.

On peut cependant désirer, lorsqu'on veut effectuer une prévision, s'affranchir de l'étape de recherche du polynôme si l'opération a déjà été effectuée une fois.

En exploitant le fichier .C, il est possible de mettre la main sur les coefficients du polynôme généré par ATLAS.

Si le tableau **variables_explicatives** contient la valeur des variables explicatives correspondant à l'événement pour lequel on veut effectuer une prévision, le programme suivant réalise cette prévision et met sa valeur dans la variable **prevision**.

```
#include "stdio.h"
#include "stdlib.h"

#include "<Préfixe><Ordre du modèle>.C"          // Chemin et nom du fichier
                                                 // source.

main(){
   float        variables_explicatives[Nombre_de_variables_en entrée] ;
```

```
    float          prevision ;

            ...          // Partie du programme initialisant variables_explicatives.

    prevision=formula<Ordre du modèle>(&variables_explicatives) ;
}
```

3. L'interprétation

L'interprétation consiste en l'analyse des résultats fournis par ATLAS. On va d'abord montrer comment atteindre ces résultats, puis comment s'en servir pour obtenir d'autres chiffres intéressants.

3.1. Récupération des résultats d'ATLAS

Les résultats de la modélisation étant contenus dans le fichier source en C, le plus simple consiste à faire cette récupération à l'intérieur d'un programme C. Pour cela, il faut inclure le fichier de sortie à l'intérieur de son propre programme via l'instruction **#include "<Préfixe><Ordre du modèle>.C"**

Rappelons ici la formule permettant d'effectuer l'estimation de la grandeur Y à l'aide des variables explicatives $X_{i,}$ produit d'un certain nombre de variables d'entrée :

$$Y = \hat{Y} + \sigma_y * \sum_i W_i \frac{X_i - \hat{X}_i}{\sigma_i}$$

Voici une liste des résultats directement accessibles.

C_NombreVE représente le nombre de variables explicatives.

moyenne [i] représente la valeur moyenne de la variable explicative X_i pour les vecteurs fournis dans le fichier d'apprentissage. C'est un tableau de type *float*.

ecartType [i] représente la valeur de l'écart type de la variable explicative X_i pour les vecteurs fournis dans le fichier d'apprentissage. C'est un tableau de type *float*.

vecteurW [i] représente la valeur du coefficient W pour la variable explicative d'indice vecteurNonNull [i]. C'est un tableau de type *float*.

Notons que ce coefficient W n'est défini que pour les variables explicatives non identiquement nulles à l'intérieur de l'espace d'apprentissage, d'où l'utilisation du tableau vecteurNonNull.

moyenne [*C_NombreVE*] représente la valeur moyenne de la grandeur à expliquer Y pour les vecteurs fournis dans le fichier d'apprentissage.

ecartType [*C_NombreVE*] représente la valeur de l'écart type de la grandeur à expliquer Y pour les vecteurs fournis dans le fichier d'apprentissage.

Reste à établir un lien entre les variables d'entrée fournies par l'utilisateur et les variables explicatives. C'est l'objet du tableau transformation.

Transformation : ce tableau permet d'expliciter les variables explicatives en fonction des variables d'entrée. C'est un tableau de type entier.

j_n = *transformation*[i+1][n] signifie que la variable explicative X_i est produit des variables d'entrées E_{j_n}

$$\text{Soit : } X_i = \prod E_{j_n}$$

3.2. Autres outils intéressants

3.2.1.Contribution :

La contribution d'une variable explicative X_i représente la part de cette variable dans le polynôme de généralisation. C'est le rapport entre la valeur absolue de son coefficient W_i et la somme des valeurs absolues de tous ces coefficients W_i

$$C_i = W_i / \sum W_j$$

Elle permet de retrouver facilement quels sont, pour un modèle donné, les coefficients prépondérants. On peut aussi suivre l'évolution de l'importance d'un coefficient avec le modèle.

3.2.2. Le rapport W_i / σ_i.

C'est le coefficient situé devant la variable x_i à l'intérieur du polynôme de généralisation (à un facteur σ_y près).

3.2.3.σ-contribution.

$C_i = (W_i / \sigma_i) / \sum (W_j / \sigma_j)$

La contribution caractérisait les variables normalisée. La σ-contribution s'occupe des variables non normalisées. Elle est intéressante lorsque l'on considère des variables booléennes, pour lesquelles l'écart-type n'apporte pas une grandeur pertinente.

3.2.4. Contribution totale

C'est la somme des contributions de toutes les variables explicatives dans lesquelles apparaît une variable d'entrée donnée. Elle caractérise donc l'importance de cette variable d'entrée.

3.2.5. Exemples.

3.2.5.1. Liste des dix contributions prépondérantes

- pour l'heure 13

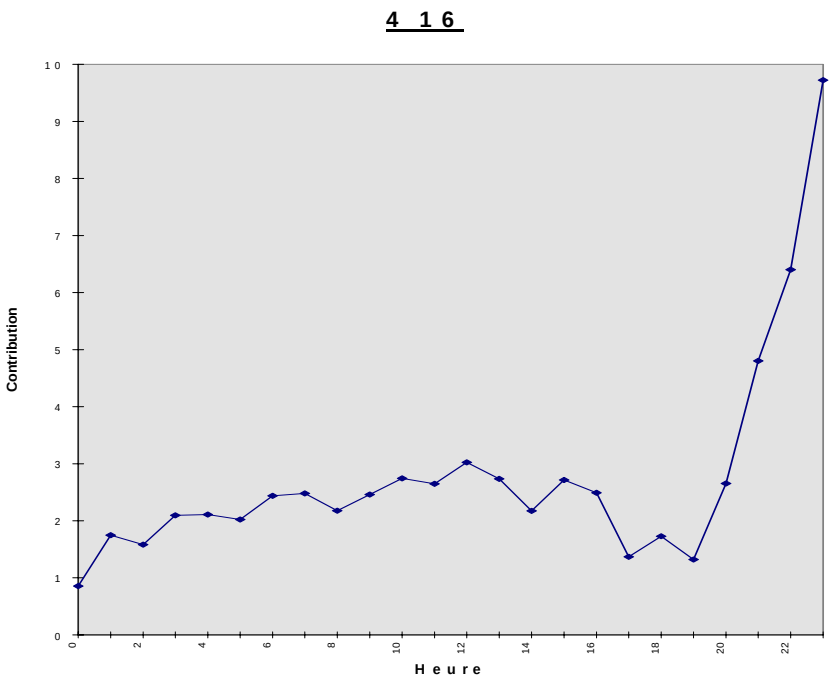
Indice	Nom	W / σ	Contribution
0	4 16	0,13	2,7
1	16 28	0,12	2,3
2	4 15	0,11	2,1
3	15 28	0,10	2,1
4	16	0,10	2,1
5	15	0,10	2,1
6	2 16	0,099	2,0
7	20	0,097	2,0
8	2 15	0,093	1,9
9	0 16	-0,088	-1,8
10	0 15	-0,085	-1,7

- pour l’heure 15

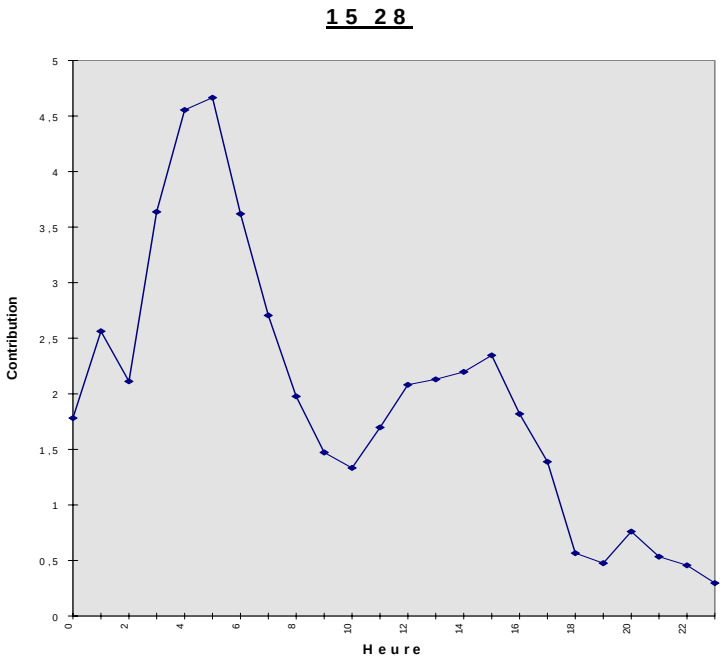
Indice	Nom	W / σ	Contribution
0	4 16	0,12	2,7
1	15 28	0,10	2,3
2	15	0,10	2,3
3	2 15	0,10	2,3
4	2 8	0,099	2,3
5	8 28	0,099	2,3
6	16 28	0,098	2,3
7	8	0,095	2,2
8	2	0,093	2,1
9	2 2	0,092	2,1
10	16	0,089	2,0

3.2.5.2.Variation des coefficients en fonction de l'heure

- coefficient 16 * coefficient 26



- coefficient 15 * coefficient 28



ANNEXE 4

Description des 29 variables explicatives.

```
case 0 :        VarExTLHebdoSin(jour,mois,annee);
case 1 :        VarExTLHebdoCos(jour,mois,annee);
case 2 :        numero_serie;
case 3 :        VarExTLMoisCos(jour);
case 4 :        VarExTLMoisSin(jour);
case 5 :        WEApresPont(numero_serie);
case 6 :        VWEApresPont(numero_serie);
case 7 :        VarExTLDebRep(numero_serie);
case 8 :        VarExTLLDebRep(numero_serie);
case 9 :        VarExTLFinRep(numero_serie);
case 10 :       VarExTLVFinRep(numero_serie);

            // variables specifique a des mois

case 11 :       debutWECongesScolaires(numero_serie);   // fevrier
case 12 :       VdebutWECongesScolaires(numero_serie);
case 13 :       LdebutWECongesScolaires(numero_serie);
case 14 :       SLdebutWECongesScolaires(numero_serie);

          // pont

case 15 :       debutWELong(numero_serie);
case 16 :       VdebutWELong(numero_serie);
case 17 :       FinWELong(numero_serie);
case 18 :       LFinWELong(numero_serie);

          // debut de conges en semaine

case 19 :       debutCongesScolaires(numero_serie);
case 20 :       LdebutCongesScolaires(numero_serie);

          // premier WE de conges scolaires

case 21 :       premierWE(numero_serie);
case 22 :       VpremierWE(numero_serie);
case 23 :       LpremierWE(numero_serie);
case 24 :       SLpremierWE(numero_serie);
case 25 :       VdebutWELong(numero_serie+1);
case 26 :       LdebutCongesScolaires(numero_serie-1);
case 27 :       VarExTLAnneeCos(jour,mois,annee);
case 28 :       VarExTLAnneeSin(jour,mois,annee);
```

Bibliographie.

Une très bonne introduction aux problèmes mal-posés est présente dans l'ouvrage de A.Tikhonov "Méthodes de résolution de problèmes mal-posés" qui reste malheureusement assez difficile à rencontrer[2]. Le reste de la bibliographie provient essentiellement de SIAM (Society for Industrial and Applied Mathematics), et notament de la parution J.Sci.Stat.Comput. qui consacre régulièrement quelques articles au problèmes mal posés.

A.Tikhonov, V.Arsénine : Méthodes de résolution de problèmes mal posés. Editions Mir, 1976.

P.C.Hansen : Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. SIAM J.Sci.Stat.Comput. Vol 11, N°3, pp 503-518, May 1990.

P.C.Hansen, D.P.O'Leary : The use of the L-curvein the regularization of discrete ill-posed problems. SIAM J.Sci.Comput. Vol 14, N°6, pp 1487-1503, November 1993.

M.R.Trummer : A method for solving ill-posed linear operator equation. SIAM J.Number.Anal. Vol 21, N°4, pp 729-737, August 1984.

L.Eldén : An algorithm for the regularization of ill-conditioned, banded least squares problems. SIAM J.Sci.Stat.Comput. Vol 5, N°1, pp 237-254 March 1984.

B.Cattan : Etude de la méthode du maximum d'entropie sur la moyenne, Application à la synthèse d'ouverture optique. Rapport d'option de l'Ecole Polytechnique, promotion 1993.

[2] Nous remercions M. G.Thauront pour nous avoir aiguillé sur cet ouvrage.